On Mixup Training: Improved Calibration and Predictive Uncertainty for Deep Neural Networks

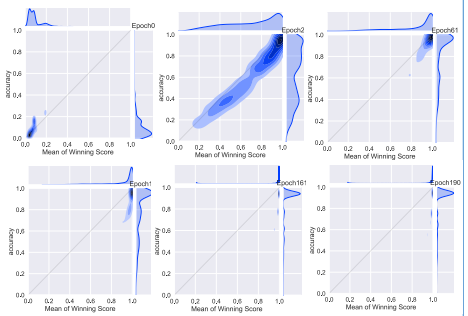
Sunil Thulasidasan[1,2], Gopinath Chennupati[1], Jeff Bilmes[2], Sarah Michalak[1], Tanmoy Bhattacharya[1]

[1]Los Alamos National Laboratory; [2]University of Washington

Los Alamos NATIONAL LABORATORY EST.1943 — ELECTRICAL & COMPUTER ENGINEERING, UNIVERSITY of WASHINGTON

Overconfidence and Uncertainty in Deep Learning

Problem: Modern DNNs are trained towards overconfidence.



Above figure shows a joint density plot of accuracy vs confidence (captured by the winning softmax score) on the CIFAR-100 validation set at different training epochs for the VGG-16 deep neural network.

In regular training, the DNN moves from under-confidence, at the beginning of training, to overconfidence at the end. A well-calibrated classifier would have most of the density lying on the $x = y$ gray line.

Towards the end of training, the DNNs are typically very overconfident i.e., the predicted scores overestimate the likelihood of correctness.

One of the factors that contribute to this is that most modern DNNs, when trained for classification in a supervised learning setting, are trained using *one-hot encoded labels that have all the probability mass in one class.*

The training labels are zero-entropy signals that always express certainty about the input. The DNN is thus, in some sense, *trained to become overconfident.*

Overview of Mixup Training

Mixup, introduced in (Zhang et al 2017) is based on the principle of vicinal risk minimization – the classifier is trained not only on the training data but also in the *vicinity* of each sample. Given two randomly selected images x_i and x_j, mixup combines them as follows:

$$\tilde{x} = \lambda x_i + (1-\lambda)x_j$$

$$\tilde{y} = \lambda y_i + (1-\lambda)y_j$$

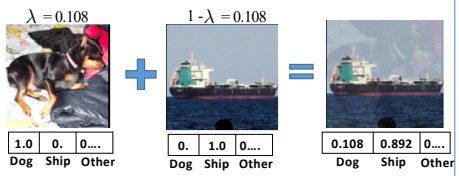
This has the effect of the empirical Dirac delta distribution

$$P_\delta(x,y) = \frac{1}{n}\sum_i^n \delta(x = x_i, y = y_i)$$

being replaced with the *empirical vicinal distribution*. The vicinal samples are generated as above, and during training minimization is performed on *empirical vicinal risk*:

$$P_\nu(\tilde{x},\tilde{y}) = \frac{1}{n}\sum_i^n \nu(\tilde{x},\tilde{y}|x_i,y_i)$$

A Mixup example for images

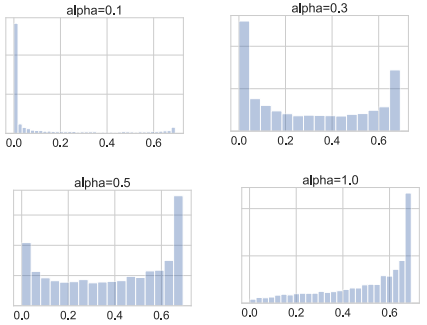


Pairs of images are convexly combined in pixel space. Depending on the mixing parameter, λ the resulting image is usually closer to one of the two images in the pairs.

But more importantly, the training labels of the original images are also convexly combined. The training label for the above mixed-up image now has probability mass in two classes.

The entropy of each original training label is zero. The entropy of the mixed-up training labels are a function of the distribution from which the mixing parameter λ is sampled. In practice, the sampling distribution is usually a symmetric Beta distribution.

Entropy Distribution of Training Labels in Mixup



Shown above is the entropy distribution of training labels as a function of the alpha parameter of the Beta(alpha, alpha) distribution from which mixing parameter is sampled.

Question: **Will this label smoothing effect of mixup lead to a better calibrated DNN?**

Since mixup produces smoothed labels over mixtures of inputs, we compare the calibration performance of mixup to two other label smoothing techniques:

- ***epsilon*-label smoothing** described in [Szegedy et al] where the one-hot encoded training signal is smoothed by distributing an epsilon mass over the other (i.e., non ground-truth) classes.
- We also compare the performance of mixup against the **entropy-regularized loss (ERL)** in [Pereyra et al] that discourages the neural network from being over-confident by penalizing low-entropy distributions.

Our baseline comparison is regular training where no label smoothing or mixing of features is applied (no-mixup).

Calibration Metrics

We measure the calibration of the network using **Expected Calibration Error.**

Let B_m be the set of samples whose prediction scores (the winning softmax score) fall into bin m. The accuracy and confidence of B_m are defined as:

$$\text{acc}(B_m) = \frac{1}{|B_m|}\sum_{i\in B_m} \mathbf{1}(\hat{y}_i = y_i)$$

$$\text{conf}(B_m) = \frac{1}{|B_m|}\sum_{i\in B_m} \hat{p}_i$$

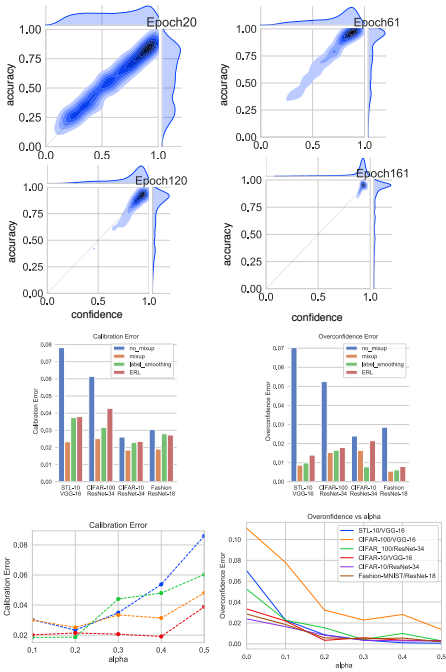
The Expected Calibration Error is then defined as

$$\text{ECE} \triangleq \sum_{m=1}^{M} \frac{|B_m|}{n}\Big|\text{acc}(B_m) - \text{conf}(B_m)\Big|$$

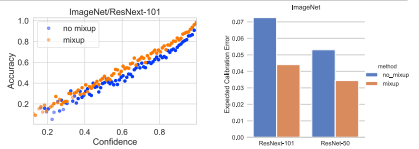
In high-risk applications, confident but wrong predictions can be especially harmful; thus we also define an additional calibration metric -- the **Overconfidence Error** (OE)-- as follows

$$\text{OE} \triangleq \sum_{m=1}^{M} \frac{|B_m|}{n}\Big(\text{conf}(B_m) \times \max\big(\text{conf}(B_m) - \text{acc}(B_m), 0\big)\Big)$$

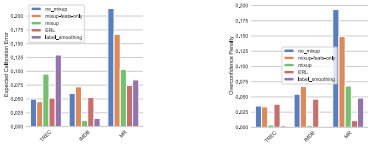
Experimental Results: Image Data



Large-Scale Experiments on ImageNet



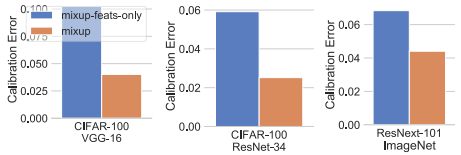
Calibration Experiments on NLP Data



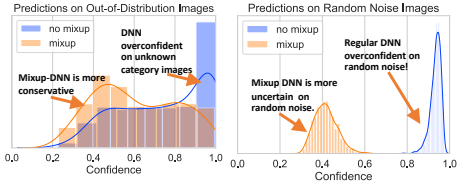
The original mixup paper worked with images and sound data. We are one of the first to extend the idea to the NLP domain, where the convex mixing is done in the *embedding layers.*

Soft Labels are Important for Calibration

Merely mixing features does not give the same calibration benefit. The label smoothing aspect of mixup has an important and beneficial effect on calibration.



Predictions on Out-of-Distribution and Random Noise Images



Conclusion: Mixup significantly improves calibration and predictive uncertainty for DNNs.

References

Thulasidasan et al On Mixup Training: *Improved Calibration and Predictive Uncertainty for Deep Neural Networks.* NeurIPS 2019

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. *mixup: Beyond empirical risk minimization.* Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. *Re- thinking the inception architecture for computer vision.* CVPR 2016

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. *On calibration of modern neural networks.* ICML 2017

Gabriel Pereyra, George Tucker, Jan Chorowski, Łukasz Kaiser, and Geoffrey Hinton. *Regularizing neural networks by penalizing confident output distributions.*

Contact: Sunil Thulasidasan (sunil@lanl.gov)

Acknowledgements: the U.S. Department of Energy, National Cancer Institute, National Institutes of Health, CONIX (DARPA)